AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for moving data objects ~~(201.x)~~ in a computer system ~~(101)~~ from a first storage location ~~(107)~~ to a second ~~(108)~~ storage location, comprising:

a) selecting one or more data objects ~~(201.x)~~ from the first storage location ~~(107)~~, ~~(301)~~[[,]];

b) assigning an identifier (ID) of a first type to each of the selected data objects ~~(201.x)~~ [[,]];

c) assigning an ID of a second type to each of the selected data objects ~~(201.x)~~ [[,]];

d) storing ~~said~~ the second type ID in a second lock object ~~(204, 302)~~ [[,]];

e) ~~in case step d) has been performed successfully (303):~~ determining whether the second type ID was stored successfully in the second lock object, and upon a successful storage, storing ~~said~~ the first type ID in a first lock object ~~(203, 307)~~ [[,]];

f) storing a data object ~~(201.x)~~, the first ID of which is contained in the first lock object ~~(203)~~, at the second storage location ~~(108)~~ ~~(608)~~ [[,]];

g) deleting [[a]] the data object ~~(201.x)~~, the first type ID of which is contained in the first lock object ~~(203)~~, from ~~said~~ the first storage location ~~(502)~~ [[,]]; and

h) deleting [[a]] the second type ID from the second lock object ~~(204) earliest at a time at which step e) for~~ after a particular first type ID has been ~~completed (308, 706)~~ stored in the first lock object.

2. (Currently Amended) The method of claim 1, further comprising:

i) deleting [[a]] the first type ID from the first lock object ~~(203) earliest at a time at which step g) for~~ the respective data object assigned to that at least one ID has been ~~completed (503)~~ deleted from the first storage location.

3. (Currently Amended) The method of claim 1 ~~or 2~~, wherein [[a]] the data object ~~(201.x)~~ comprises one ~~or~~ or more fields of one or more tables, ~~(201, 202)~~ and wherein the at least one ID comprises one or more key fields of the one or more tables.

4. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 3~~, wherein ~~in step f)~~ the data ~~objects (201.x) are~~ object is stored in one or more files and wherein an assignment of a first type ID to the file or to a name of the file, in which the data object assigned to ~~said~~ the first type ID is stored, is stored in the first lock object ~~(203) (609)~~.

5. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 4~~, wherein the first lock object ~~(203)~~ is stored on a nonvolatile storage means.

6. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 5~~, wherein ~~in step d)~~ the second type ID is stored in the second lock ~~(204)~~ object ~~immediately~~ after

~~performing step c)~~ the step of assigning an ID of a second type to each of the selected data objects for the respective data object ~~(201.x)~~.

7. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 5~~, wherein ~~in step d)~~ the second type of ID of the selected data object ~~(201.x)~~ is stored in the second lock object ~~(204) shortly~~ before ~~the storing process according to step f) for~~ the data object ~~(201.x)~~ assigned to that ID is stored ~~started~~.

8. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 7~~, wherein storing the first type ID in a first lock object further comprises:

~~in step e)~~ storing the first type IDs of all selected data objects ~~(201.x) are stored~~ before ~~the~~ a first storing process according to step f) is started.

9. (Currently Amended) The method ~~one~~ of ~~claims~~ claim 1 ~~to 8~~, further comprising:

j) checking ~~before or while performing any of steps a) to e) for a data object (201.x),~~ whether a first type ID for the data object has been stored in a first lock object ~~(203)~~, and if ~~yes~~ the first type ID for that data object has been stored, skipping at least step f) for that data object ~~(201.x)~~.

10. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 8~~, further comprising:

k) checking ~~before or while performing any of steps a) to f) for a data object (201.x),~~ whether ~~that~~ the data object is contained in the second storage location ~~(108)~~,

and if the data object is contained in the second storage location ~~yes~~, skipping at least step f) for that data object ~~(201.x)~~.

11. (Currently Amended) The method of claim 10, wherein ~~said~~ the checking step k) is performed by querying [[a]] the first lock object ~~(203)~~.

12. (Currently Amended) The method of ~~one of claims 2 to 11~~ claim 1, further comprising:

l) ~~in case of a failure in step f)~~ determining whether the data object was stored in the first lock object successfully, and upon a successful storage, checking[[,]] whether the data object ~~(201.x)~~ assigned to the respective first ID has been completely stored in the second storage location ~~(108)~~, and ~~in case of no~~ if the respective first ID has not been stored, skipping at least steps g) and h) for that data object ~~(201.x)~~ and deleting the first ID from the first lock object ~~(203)~~.

13. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 12~~ for use in an enterprise resource planning software.

14. (Currently Amended) A computer system for processing data ~~by means of or in a software application~~, comprising:

[[-]]memory means for storing program instructions;

[[-]]input means for entering data;

[[-]]storage means for storing data;

[[-]]a processor responsive to the program instructions, wherein the

[[-]]program instructions ~~to carry out a method as of any of claims 1 to 12~~ comprise program code means for performing a method for moving data objects in the computer system from a first storage location to a second storage location, the method comprising:

selecting one or more data objects from the first storage location;

assigning an identifier (ID) of a first type to each of the selected data objects;

assigning an ID of a second type to each of the selected data objects;

storing the second type ID in a second lock object;

determining whether the second type ID was stored successfully in the second lock object, and upon a successful storage, storing the first type ID in a first lock object;

storing a data object, the first ID of which is contained in the first lock object, at the second storage location;

deleting the data object, the first type ID of which is contained in the first lock object, from the first storage location; and

deleting the second type ID from the second lock object after a particular first type ID has been stored in the first lock object.

15. (Cancelled)

16. (Currently Amended) A computer readable medium comprising ~~program code~~ instructions for performing a method ~~as of any of claims 1 to 13 if said program code is executed on a computer system~~ for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting one or more data objects from the first storage location;

assigning an identifier (ID) of a first type to each of the selected data objects;

assigning an ID of a second type to each of the selected data objects;

storing the second type ID in a second lock object;

determining whether the second type ID was stored successfully in the second lock object, and upon a successful storage, storing the first type ID in a first lock object;

storing a data object, the first ID of which is contained in the first lock object, at the second storage location;

deleting the data object, the first type ID of which is contained in the first lock object, from the first storage location; and

deleting the second type ID from the second lock object after a particular first type ID has been stored in the first lock object.

17. (Cancelled)

18. (New) The computer readable medium of claim 16, further comprising:

deleting the first type ID from the first lock object the respective data object assigned to that at least one ID has been deleted from the first storage location.

19. (New) The computer readable medium of claim 16, wherein

the data object comprises one or more fields of one or more tables, and wherein the at least one ID comprises one or more key fields of the one or more tables.

20. (New) The computer readable medium of claim 16, wherein

the data object is stored in one or more files and wherein an assignment of a first type ID to the file or to a name of the file, in which the data object assigned to the first type ID is stored, is stored in the first lock object.

21. (New) The computer readable medium of claim 16, wherein the first lock object is stored on a nonvolatile storage means.

22. (New) The computer readable medium of claim 16, wherein the second type ID is stored in the second lock object after the step of assigning an ID of a second type to each of the selected data objects for the respective data object.

23. (New) The computer readable medium of claim 16, wherein

the second type of ID of the selected data object is stored in the second lock object before the data object assigned to that ID is stored.

24. (New) The computer readable medium of claim 16, wherein storing the first type ID in a first lock object further comprises:

storing the first type IDs of all selected data objects before storing the data object at the second storage location.

25. (New) The computer readable medium of claim 16, further comprising:

checking whether a first type ID for the data object has been stored in a first lock object, and if the first type ID for that data object has been stored, skipping the storing the data object, the first ID of which is contained in the first lock object at the second storage location, for that data object.

26. (New) The computer readable medium of claim 16, further comprising:

checking whether that data object is contained in the second storage location, and if the data object is contained in the second storage location, skipping the storing the data object, the first ID of which is contained in the first lock object at the second storage location, for that data object.

27. (New) The computer readable medium of claim 16, wherein the checking is performed by querying the first lock object.

28. (New) The computer readable medium of claim 16, further comprising:

determining whether the data object was stored in the first lock object successfully, and upon a successful storage, checking whether the data object assigned to the respective first ID has been completely stored in the second storage location, and if the respective first ID has not been stored, skipping the deleting the data object from the first storage location and the deleting the second type ID from the second lock object after a particular first type ID has been stored in the first lock object for that data object and deleting the first ID from the first lock object.

29. (New) A computerized system for processing data, comprising:

selecting one or more data objects from the first storage location;

assigning an identifier (ID) of a first type to each of the selected data objects;

assigning an ID of a second type to each of the selected data objects;

storing the second type ID in a second lock object;

determining whether the second type ID was stored successfully in the second lock object, and upon a successful storage, storing the first type ID in a first lock object;

storing a data object, the first ID of which is contained in the first lock object, at the second storage location;

deleting the data object, the first type ID of which is contained in the first lock object, from the first storage location; and

deleting the second type ID from the second lock object after a particular first type ID has been stored in the first lock object.

30. (New) The computerized system of claim 29, further comprising:

deleting the first type ID from the first lock object the respective data object assigned to that at least one ID has been deleted from the first storage location.

31. (New) The computerized system of claim 29, wherein the data object comprises one or more fields of one or more tables, and wherein the at least one ID comprises one or more key fields of the one or more tables.

32. (New) The computerized system of claim 29, wherein

the data object is stored in one or more files and wherein an assignment of a first type ID to the file or to a name of the file, in which the data object assigned to the first type ID is stored, is stored in the first lock object.

33. (New) The computerized system of claim 29, wherein the first lock object is stored on a nonvolatile storage means.

34. (New) The computerized system of claim 29, wherein the second type ID is stored in the second lock object after the step of assigning an ID of a second type to each of the selected data objects for the respective data object.

35. (New) The computerized system of claim 29, wherein

the second type of ID of the selected data object is stored in the second lock object before the data object assigned to that ID is stored.

36. (New) The computerized system of claim 29, wherein storing the first type ID in a first lock object further comprises:

storing the first type IDs of all selected data objects before storing the data object at the second storage location.

37. (New) The computerized system of claim 29, further comprising:

checking whether a first type ID for the data object has been stored in a first lock object, and if the first type ID for that data object has been stored, skipping the storing the data object, the first ID of which is contained in the first lock object at the second storage location, for that data object.

38. (New) The computerized system of claim 29, further comprising:

checking whether the data object is contained in the second storage location, and if the data object is contained in the second storage location, skipping the storing

the data object, the first ID of which is contained in the first lock object at the second storage location, for that data object.

39. (New) The computerized system of claim 29, wherein the checking is performed by querying the first lock object.

40. (New) The computerized system of claim 29, further comprising:

determining whether the data object was stored in the first lock object successfully, and upon a successful storage, checking whether the data object assigned to the respective first ID has been completely stored in the second storage location, and if the respective first ID has not been stored, skipping the deleting the data object from the first storage location and the deleting the second type ID from the second lock object after a particular first type ID has been stored in the first lock object for that data object and deleting the first ID from the first lock object.